

ADAU1466SOM – miniaturowy moduł DSP (3)

Aplikacja testowa

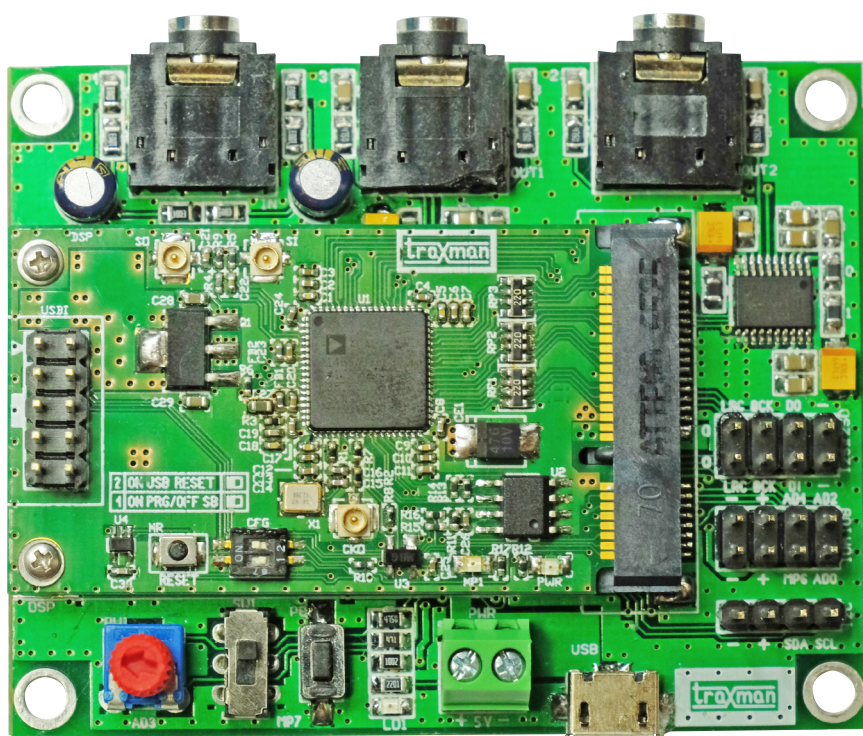
W poprzednich częściach artykułu opisano moduł ADAU1466SOM wraz ze współpracującą płytką bazową ADAU1466_SOM_MB. Skoro sprzęt jest już gotowy do działania, to nadeszła kolej na wprowadzenie do konfiguracji i oprogramowania układu w celu jego zastosowania w praktyce.

Przed rozpoczęciem testowania konieczne jest przygotowanie zestawu płytki bazowej, zasilacza, kabli UFL-RCA (2 szt.), stereofonicznego mini jacka 3,5 mm (3 szt.) oraz źródła, np. odtwarzacza CD z wyjściem S/PDIF (coax), przetwornika C/A z wejściem S/PDIF (coax) oraz wzmacniacza audio. Zestaw – podobnie jak wcześniejsze procesory z rodziny Sigma DSP – jest programowany i konfigurowany za pomocą programatora USBi. Moduł SOM wraz z płytką bazową pokazano na **fotografii 1**, natomiast programator USBi na **fotografii 2**. Dla przypomnienia, na **rysunku 3** zamieszczono schemat blokowy układu ADAU1466.

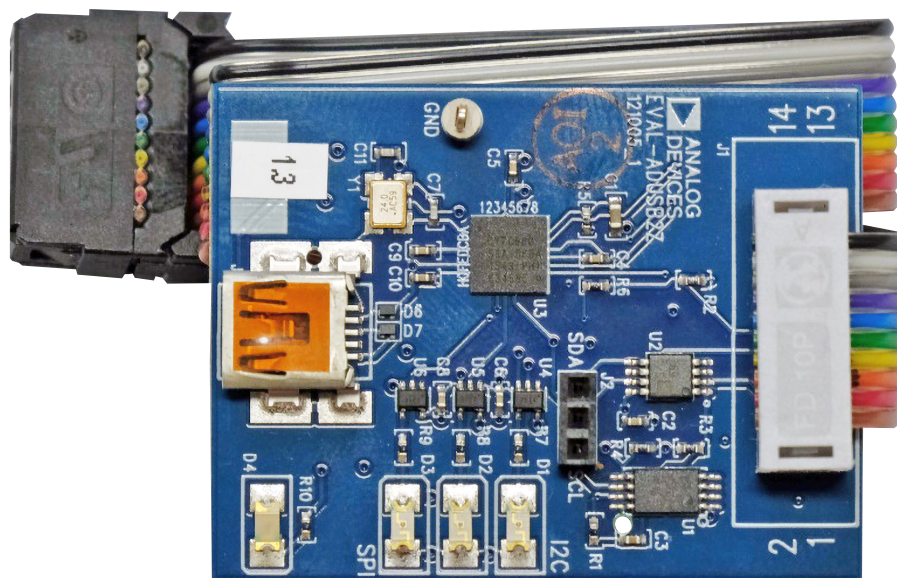
Ze strony <https://www.analog.com> należy pobrać oprogramowanie SigmaDSP (rysunek 4). W momencie pisania artykułu posługiwałem się wersją 4.1. Do jego pracy jest wymagany komputer PC z zainstalowanym systemem Windows 7 lub nowszym oraz interfejsem USB 2.0. Po zainstalowaniu i uruchomieniu należy dołączyć programator USBi do komputera i płytki ADAU1466SOM oraz założyć i zapisać nowy projekt.

Uruchomienie aplikacji przykładowej dla DSP składa się z kilku kroków:

- **Konfiguracja częstotliwości próbkowania** systemu z menu (lub CTRL+U, CTRL+Q), jak na **rysunku 5** oraz zatwierdzenie zmian klawiszem, co spowoduje parametryzację użytych bloków funkcjonalnych do ustalonej częstotliwości próbkowania.
- **Konfiguracja interfejsu programowania USBi** (rysunek 6). Procesor DSP jest programowany przez interfejs USBi



Fotografia 1. Moduł ADAU1466_SOM_I2C wraz z płytką bazową

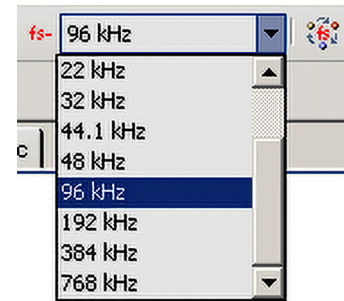


Fotografia 2. Programator USBi

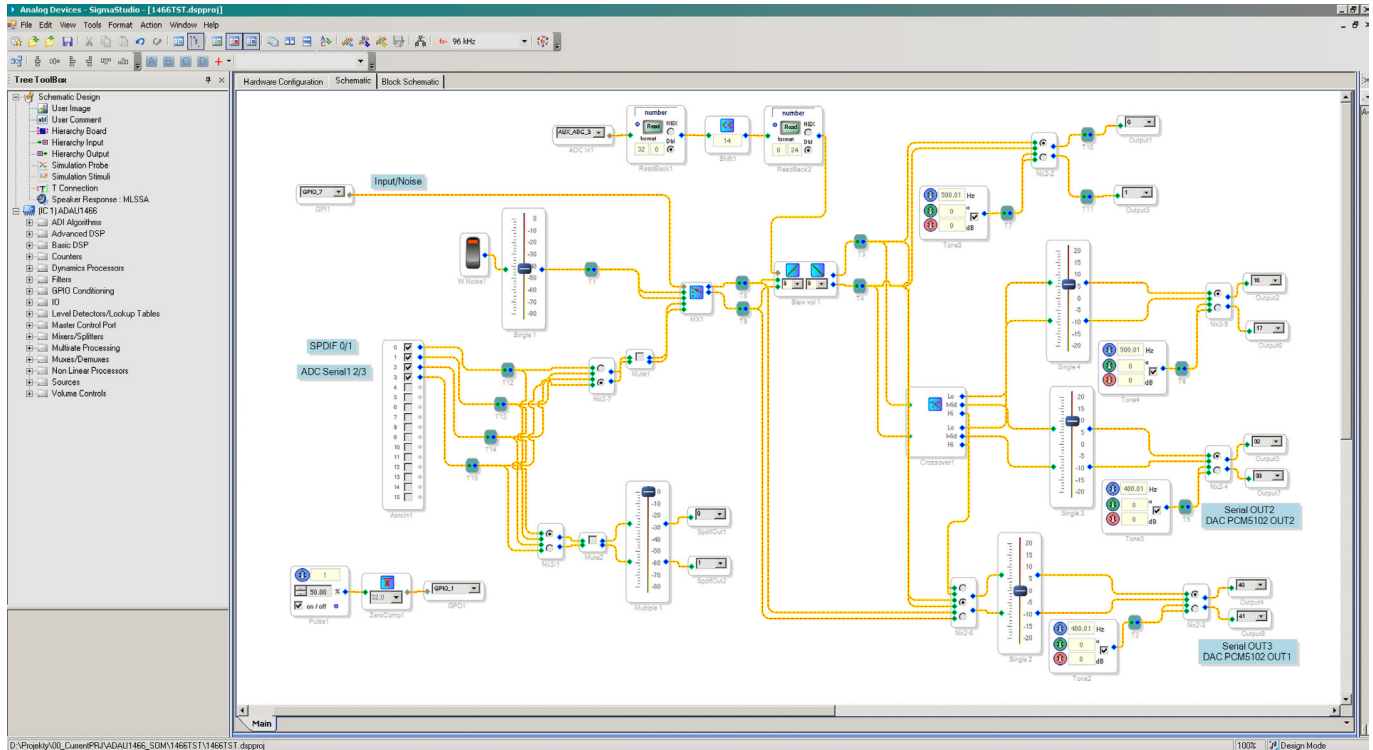
(interfejs slave w DSP) przy użyciu interfejsu I²C. Zgłasza się on pod adresem 0x70. Z procesorem współpracuje pamięć programu I²C (interfejs master w DSP) pracująca pod adresem 0xA0. **Należy uważnie sprawdzić wpisane adresy, aby nie przeprogramować firmware'u interfejsu USBi, co zakończy się jego trwałym uszkodzeniem!** W tym kroku konieczne jest także ustawianie konfiguracji pamięci EEPROM (rysunek 7).

Okienko konfiguracji dostępne jest po kliknięciu prawym przyciskiem myszy na układ U2 E2prom. Ustawione parametry są dopasowane do pamięci AT24C512, jeżeli zastosowano pamięć o innej konfiguracji, trzeba je ustawić zgodnie z notą producenta.

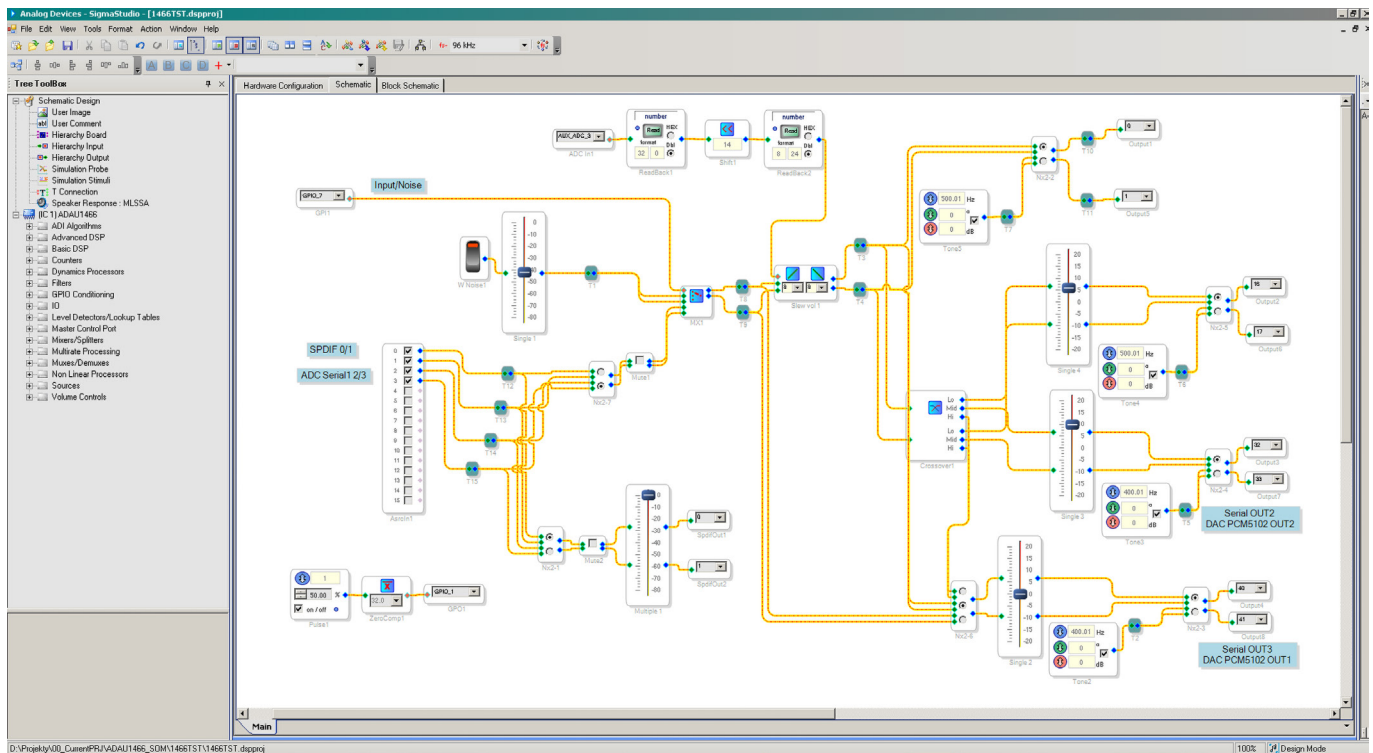
- **Konfiguracji procesora ADAU1466** poprzez ustawienie odpowiednich opcji w menu „Hardware Configuration” w zakładce IC1-ADAU1466 „Register



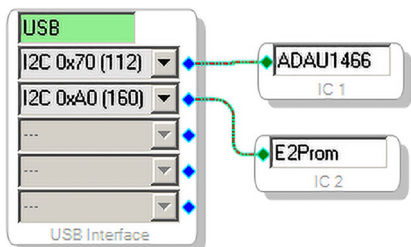
Rysunek 5. Konfiguracja częstotliwości fs



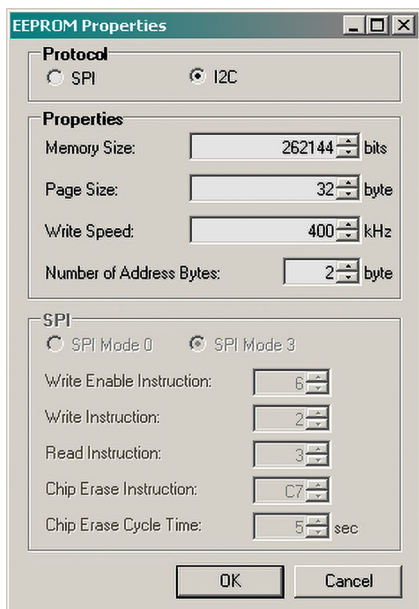
Rysunek 3. Schemat blokowy ADAU1466



Rysunek 4. Uruchomione Sigma Studio 4.1



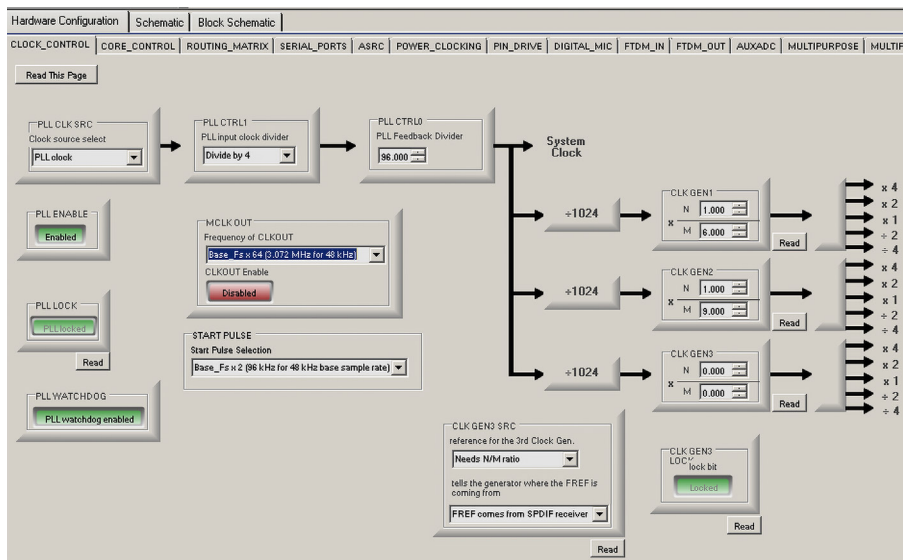
Rysunek 6. Konfiguracja USBi



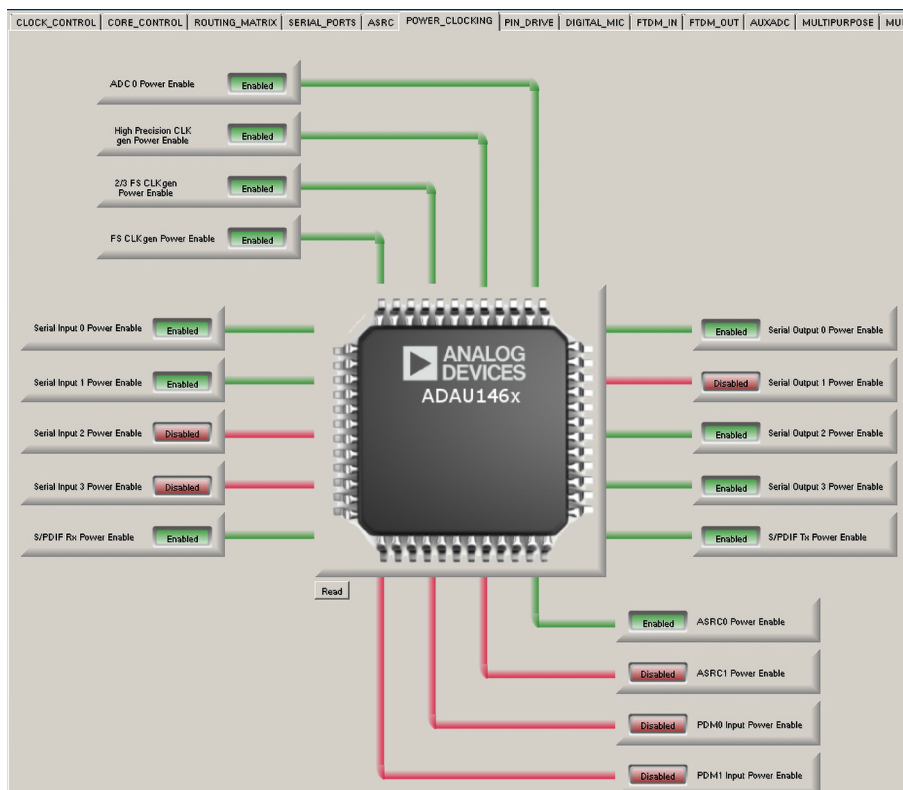
Rysunek 7. Konfiguracja pamięci EEPROM

Controls”. Kompletna konfiguracja wymaga ustawienia sporej liczby parametrów poukrywanych w zakładkach. Na szczęście w nowej wersji są one logicznie pogrupowane:

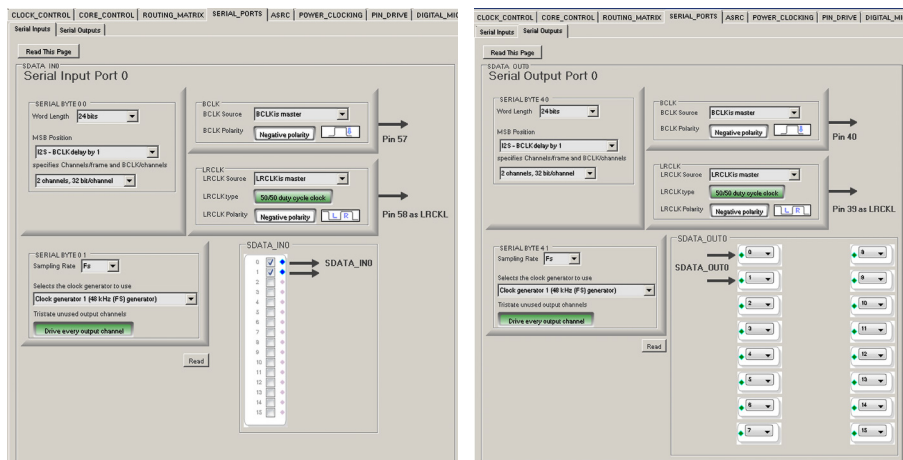
- *Clock_Control* odpowiada za konfigurację systemu taktowania DSP oraz konfigurację generatora sygnału MCLK (**rysunek 8**).
- *Power_Clocking*, gdzie jest konfigurowane zasilanie poszczególnych układów DSP. Polecam załączanie zasilania tylko bloków wymaganych w projekcie dla zaoszczędzenia pobieranej mocy i minimalizacji zaburzeń EMI (**rysunek 9**).
- *Serial_ports*, (*Input/Output*), gdzie konfigurowane są interfejsy szeregowy, określane są tryby ich pracy (master, slave), rodzaj interfejsu (I²S, TDM, LJ, RJ), długość słowa danych od 16 do 32 bitów (**rysunek 10**).
- *SPDIF*, kilka zakładek, w których są konfigurowane interfejsy S/PDIF RX/TX (**rysunek 11**).
- *Multipurpose* określająca konfigurację GPIO DSP. Ze względu na kilka funkcji pełnionych przez GPIO, określane w niej jest kierunek pracy GPIO (In/Out), czas eliminacji drgań (Debounce) w wypadku wejścia, tryb podciągania w przypadku wyjścia oraz multipleksowanie wyprowadzenia pomiędzy



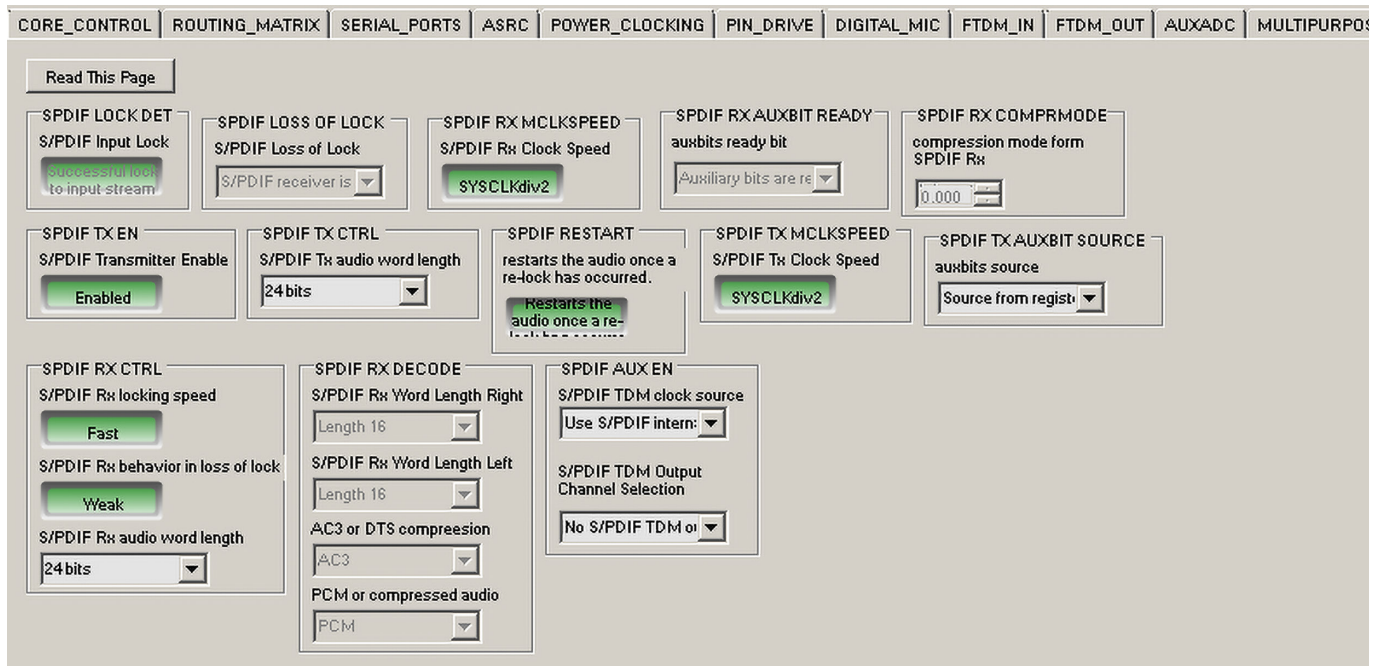
Rysunek 8. Konfiguracja taktowania procesora *Clock_Control*



Rysunek 9. Konfiguracja zasilania *Power_Clocking*



Rysunek 10. Konfiguracja interfejsów szeregowych *Serial_ports*

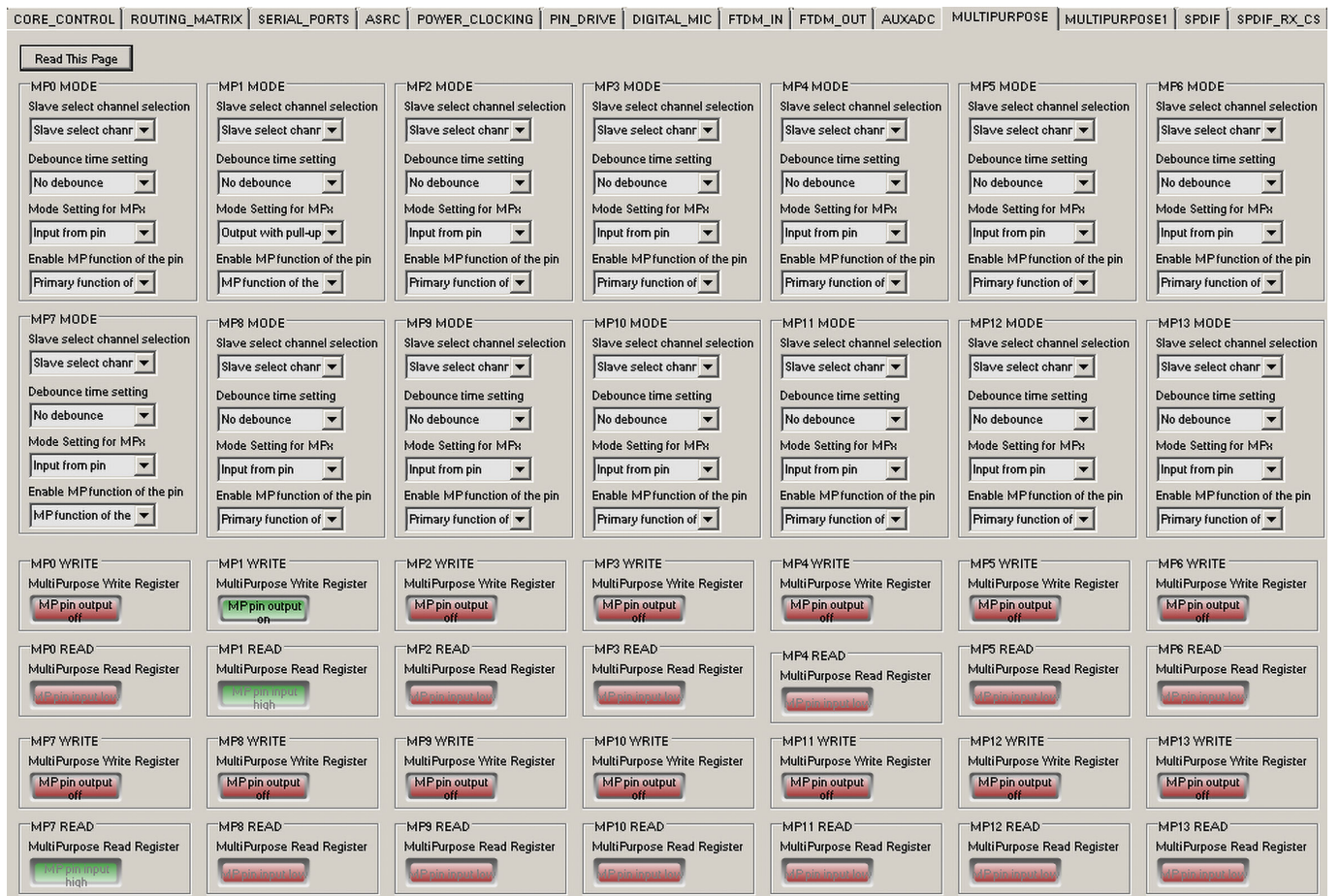


Rysunek 11. Konfiguracja interfejsów SPDIF

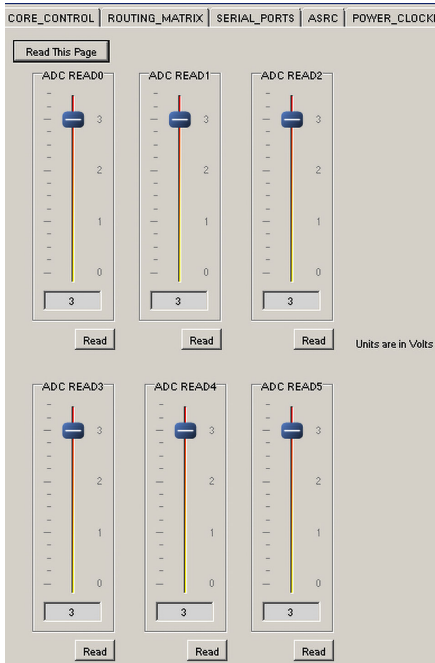
GPIO a rdzeń DSP, gdy może ono być skonfigurowane jako interfejs szeregowy lub interfejs pamięci zewnętrznej. Za pomocą funkcji *Read This Page* jest możliwe odczytanie stanu wejść GPIO, a za pomocą przycisków funkcyjnych *MPxx Write* – zmiana poziomu wyjść (rysunek 12).

– *AUXADC*, zakładka tylko do odczytu poziomów napięć wbudowanego przetwornika A/C. Jej działanie jest zastanawiające, ponieważ poziomy zmieniają się skokowo 0-3, niezależnie od wartości napięcia na A/C. Błąd ten jest zgłoszony producentowi i zapewne zostanie naprawiony

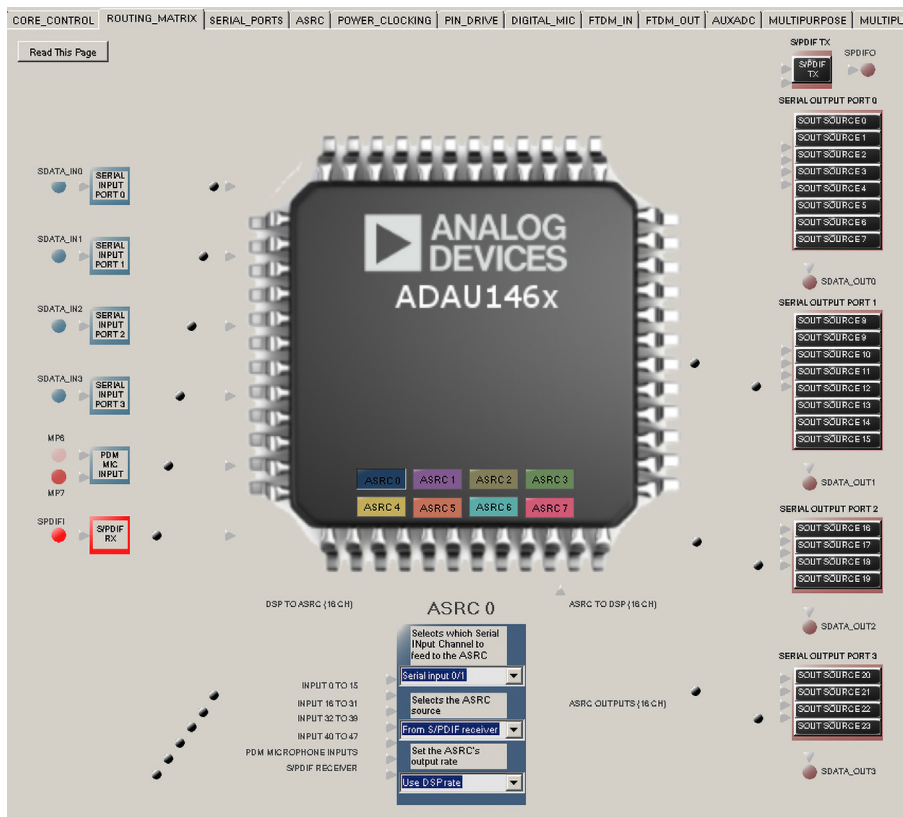
w którejś z kolejnych wersji Sigma Studio (rysunek 13).
 – *Routing Matrix*, zakładka określająca sposób przepływu sygnałów audio przez DSP i ASRC (konwerter częstotliwości próbkowania) w aplikacjach wymagających zmiany częstotliwości próbkowania do fs DSP (rysunek 14).



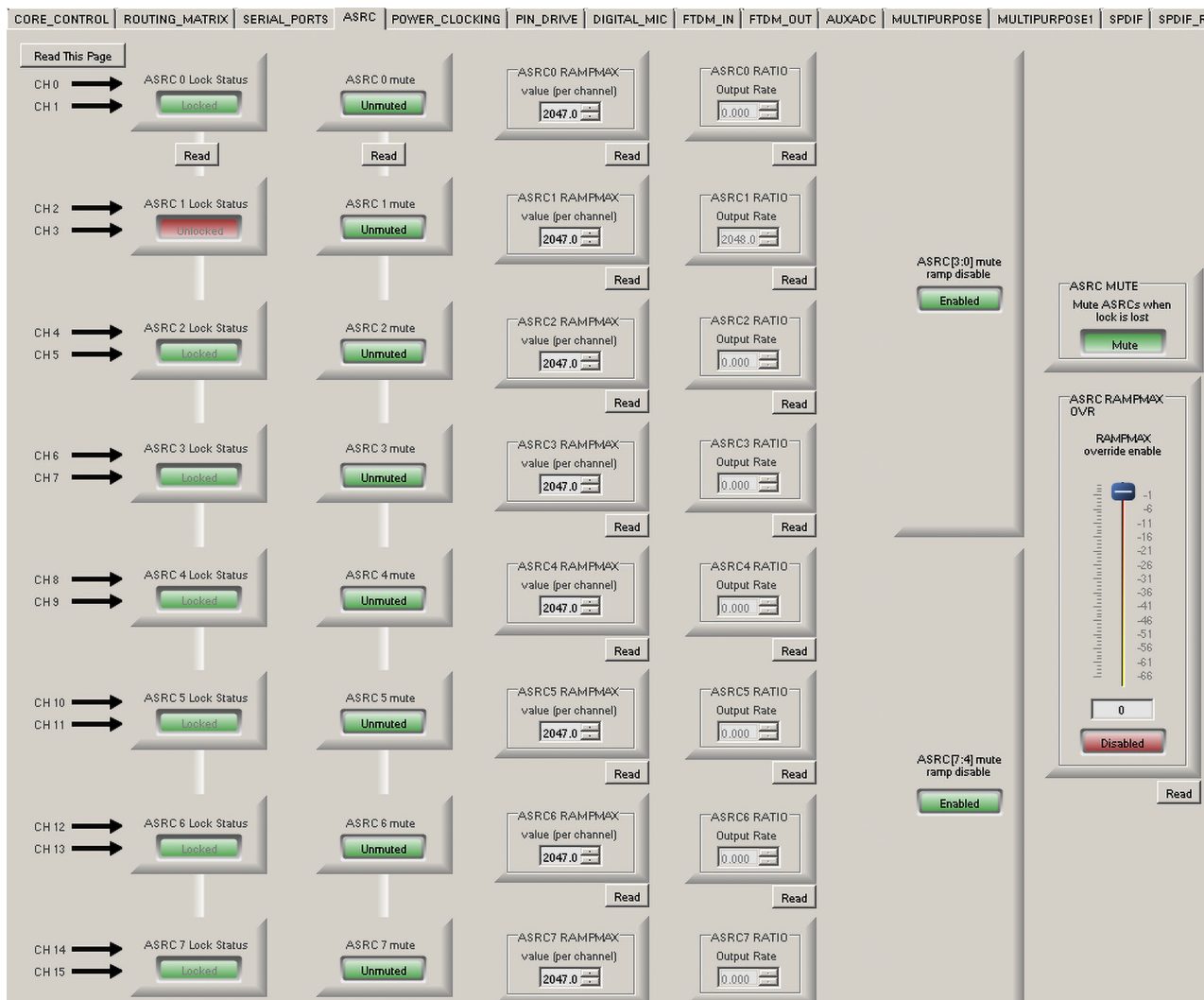
Rysunek 12. Konfiguracja interfejsów GPIO



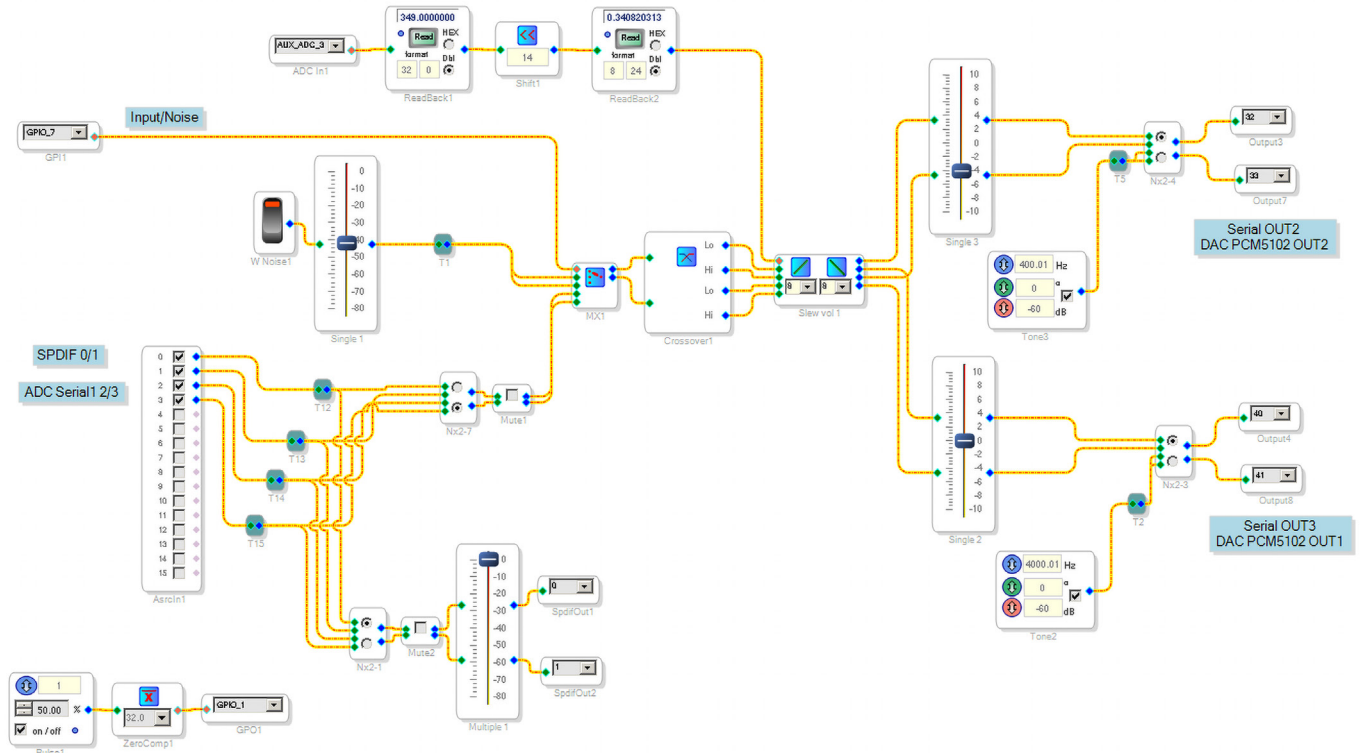
Rysunek 13. Konfiguracja przetwornika AUXADC



Rysunek 14. Konfiguracja przepływu sygnałów Routing Matrix



Rysunek 15. Zakładka monitorująca stan ASRC



Rysunek 16. Przykładowa aplikacja

– ASRC zakładka informująca o stanie układu ASRC (rysunek 15).

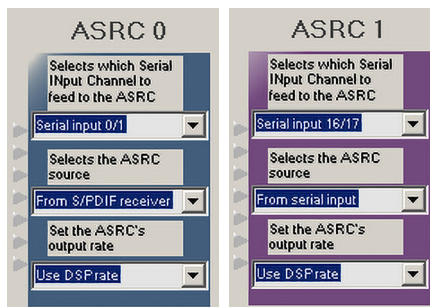
Jak widać, parametrów do ustawiania jest sporo, co wymaga częstego sięgania do dosyć obszernej dokumentacji lub pomocy na forum ez.analog. Niestety, od kilku lat nie można doczekać się sensownego zestawu aplikacji przykładowych, ułatwiających start z procesorami SigmaDSP. O ile w wypadku w miarę nieskomplikowanych ADAU1701 można poradzić sobie całkiem szybko, to konfigurowanie ADAU1466 jest czasochłonne.

W kolejnym kroku należy narysować aplikację – nieskomplikowaną, korzystając z zakładki *Schematic*, a bardziej złożoną

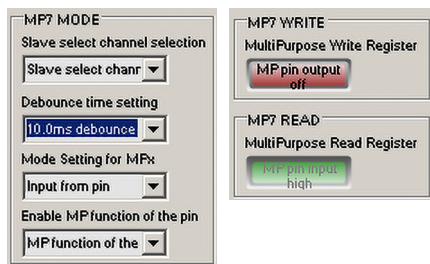
– z zakładki schematu blokowego *Block Schematic*. Aplikacja jest tworzona z gotowych bloków funkcjonalnych dostępnych w bibliotekach programu. Łączone są one za pomocą „przewodów”, co upodabnia aplikacje do rzeczywistego układu złożonego z połączonych fizycznych urządzeń. Pozwala

to na skupienie się na realizacji algorytmu, a nie na języku programowania.

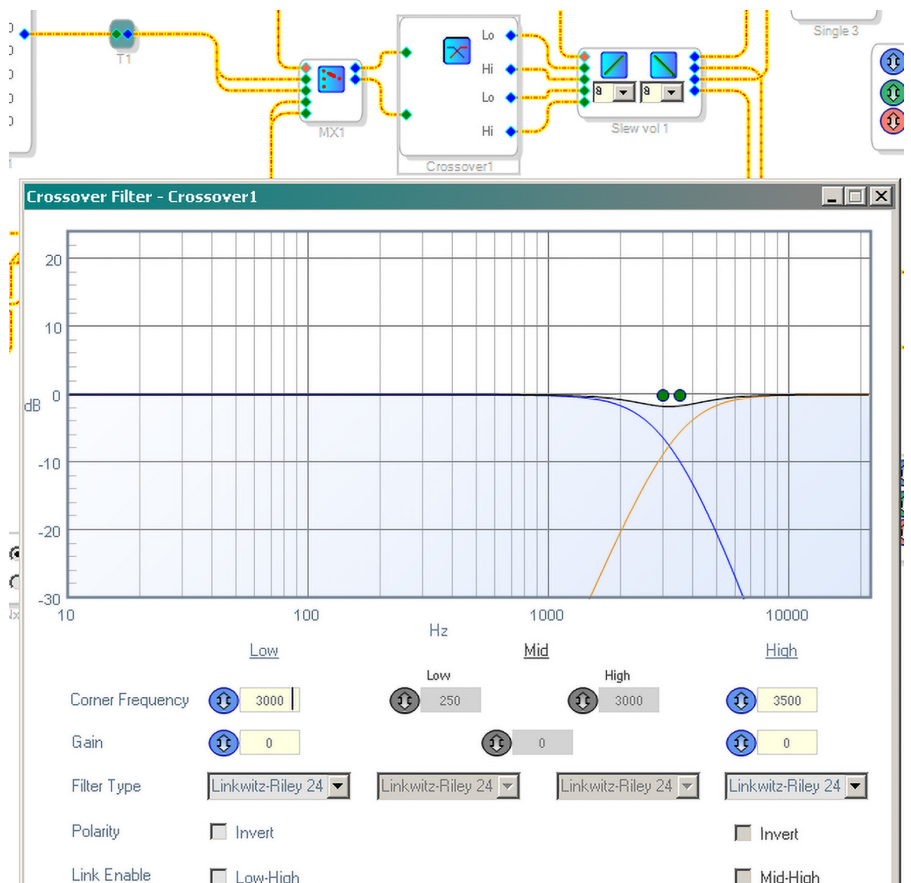
Edytor ma wyraźne oznaki choroby wieku niemowlęcego. Ma kilka uciążliwych przypadłości, do których na razie trzeba się przyzwyczaić. Na przykład, niekiedy błędnie odświeża schemat, co skutkuje jego



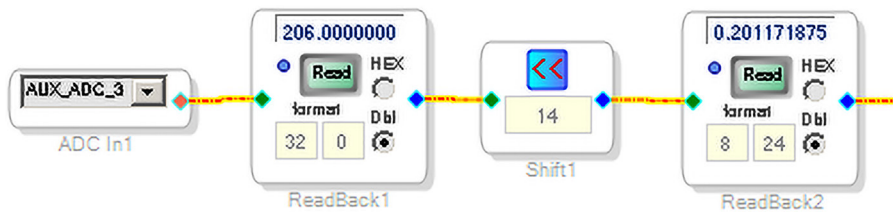
Rysunek 17. Konfiguracja źródeł ASRC



Rysunek 18. Konfiguracja GPIO7(MP7)



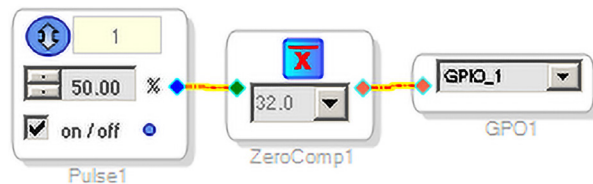
Rysunek 19. Konfiguracja bloku zwrotnicy



Rysunek 20. Konwersja formatu sygnału A/C

zamrożeniem (nie można wpisać wartości w konfiguracji bloków). Czasem IDE pracuje niestabilnie, a najbardziej uciążliwy jest jednak brak poprawnego działania funkcji kopiuj/wklej. Po takiej operacji na zaznaczonych elementach wklejane są czasem jako grafika, co skutecznie uniemożliwia podłączenie ich do „przewodów” i w miarę sprawne rysowanie schematu. W razie konieczności powielenia fragmentu schematu łatwiej jest narysować go ponownie, niż skopiować. Na pozostałe dolegliwości pomaga restart środowiska. Warto też wyrobić sobie nawyk zapisywania co pewien czas projektu za pomocą naciśnięcia klawiszy Ctrl+S, co ustrzeże przed niezamierzoną utratą pracy.

Przykładową aplikację wykonaną za pomocą Sigma Studio pokazano na **rysunku 16**. Pełni ona funkcję stereofonicznej zwrotnicy dwudrożnej wykonanej do celów dydaktycznych. Sygnał wejściowy z interfejsu S/PDIF oraz z portu Serial1, do którego jest doprowadzony przetwornik A/C przez interfejs I²S (A/C w trybie master, tj. generuje sygnały



Rysunek 21. Wskaźnik pracy DSP

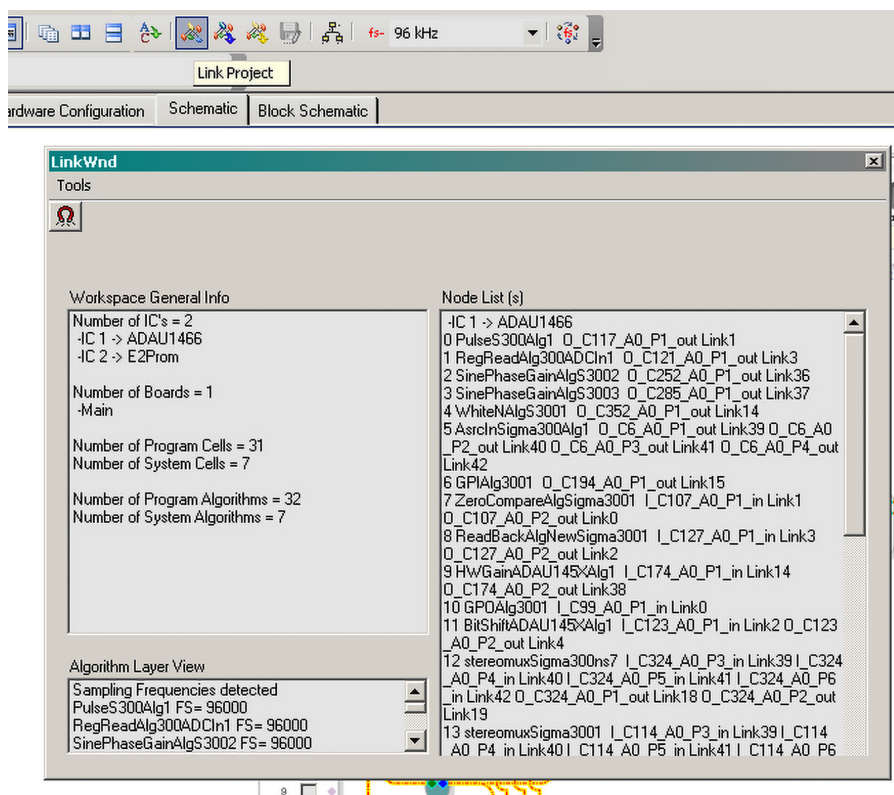
LRCK/BCLK), jest doprowadzony do bloku ASRC. Blok konwersji częstotliwości próbkowania ASRC przelicza mogące zmieniać się częstotliwości i rozdzielczości próbkowania sygnałów z interfejsów wejściowych oraz synchronizuje je z rdzeniem DSP. Jest to preferowany sposób doprowadzenia sygnałów do obróbki DSP, chyba że mamy absolutną pewność, że sygnały interfejsów szeregowych nie będą zmieniały częstotliwości próbkowania fs oraz są zgodne z częstotliwością próbkowania DSP. Dla interfejsu S/PDIF (kanały 0/1) ASRC jest obligatoryjne. Nieco myląca jest nazwa bloku AsrcIn, co sugeruje wejście do ASRC. W rzeczywistości założono, że oznaczenie In odpowiada kierunkowi sygnałów widzianych przez DSP (In – sygnał do obróbki doprowadzony do rdzenia DSP). Konfigurację ASRC (zakładka *Routing Matrix*) pokazano na **rysunku 17**.

Procesor ADAU1466 ma 8 niezależnych bloków ASRC (stereofoniczne lub dwa kanały w trybie multipleksowanym TDM).

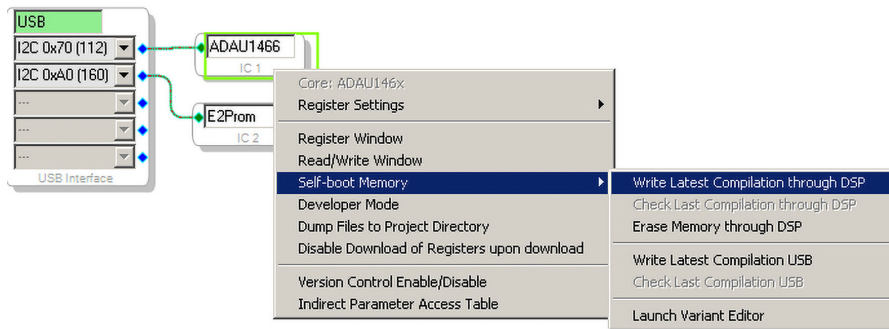
Z bloku ASRC sygnał poprzez punkty połączeniowe T-Connection jest doprowadzony do dwóch multiplekserów. Pierwszy pozwala na przełączenie sygnału z interfejsu S/PDIF-Serial1 do dalszej obróbki, drugi doprowadza go do regulatora poziomu i na wyjście S/PDIF. W tor sygnału są włączone bloki wyciszenia. Blok Mute i blok multipleksersa są „sterowane” z aplikacji Sigma Studio (za pomocą myszki) i nie mają możliwości zmiany stanu poprzez interfejs GPIO/AUXADC. Są jednak bardzo użyteczne podczas uruchamiania aplikacji. Dalej, w torze DSP, sygnał jest podany na multiplekser MX1, ale w wersji ze sterowaniem zewnętrznym (z GPIO lub z innego bloku funkcjonalnego) – w naszym wypadku z wejścia GPIO_7 (MP7). Zmiana poziomu MP7 pozwala na przełączanie multipleksersa pomiędzy generatorem szumu a sygnałem audio z ASRC. Aby możliwa była taka interakcja, konieczne jest skonfigurowanie GPIO7 w tryb wejścia i załączenie funkcji MP na pinie GPIO7 (**rysunek 18**). W modelu do GPIO7 są dołączone przełącznik suwakowy i przycisk, które po załadowaniu aplikacji będą aktywowały generator szumu „W_Noise”. Poziom szumu może być ustalony regulatorem podczas uruchamiania aplikacji. Następnie sygnał jest podany na blok zwrotnicy dwudrożnej „Crossover1”, której parametry można ustalić, klikając lewym przyciskiem myszy na ikonę zwrotnicy (**rysunek 19**).

Sygnał po podziale na składowe o częstotliwości niskiej lub wysokiej jest doprowadzony do bloku regulatora poziomu sterowanego sygnałem zewnętrznym Slew vol, w którym należy zwiększyć liczbę kanałów z 2 do 4 za pomocą menu podręcznego (prawy przycisk myszy → *Grow Algorithm* → *Gain (HW slew)* → 2. Analogicznie zmniejsza się liczbę kanałów (prawy przycisk myszy → *Reduce Algorithm*), o ile zajdzie taka potrzeba i dany blok umożliwia operację. Ze względu na zmianę formatu danych w ADAU1466 (w porównaniu do ADAU1701), regulacja poziomu sygnałem z przetwornika A/C nie może odbywać się bezpośrednio, a dopiero po konwersji formatu za pomocą dodatkowych blozków, zgodnie ze schematem zamieszczonym na **rysunku 20**.

Na koniec sygnał jest doprowadzony do bloków korygujących poziom już po podziale częstotliwości i wyprowadzony na wyjścia portu szeregowego współpracujące z przetwornikami C/A. Dla celów testowych wyjścia mogą być przełączone multipleksersami na sygnał z testowych generatorów przebiegu sinusoidalnego 400 Hz/4 kHz. Ostatnim blokiem aplikacji jest blok sygnalizujący pracę procesora DSP (**rysunek 21**). Generator impulsów o częstotliwości 1 Hz i wypełnieniu



Rysunek 22. Aplikacja po linkowaniu



Rysunek 23. Programowanie pamięci EEPROM

50% jest połączony z blokiem komparatora wykrywającego poziom niski, a stąd doprowadzony do wyjścia GPIO1 dołączonego do LED MP1.

Po narysowaniu schematu i konfiguracji sprzętu, ustawieniu przełączników CFG na płytce w położenie 1=ON, 2=ON, co dezaktywuje tryb selfboot, dołącza sygnał reset USBi do systemu i umożliwia programowanie DSP. W tym momencie należy załadować aplikację do ADAU1466. Wymaga ona linkowania (Ctrl+I, rysunek 22) i kompilacji (F7), jeżeli ładowanie zakończy się poprawnie, to na belce statusu zostanie wyświetlony komunikat. Aplikacja zostanie uruchomiona, co potwierdza miganie Led

MP1. Przechodzimy do sprawdzenia działania aplikacji w systemie. Parametry bloków funkcjonalnych mogą być zmieniane w czasie rzeczywistym, ponieważ bezpośrednio przez USBi ładowane są do pamięci DSP.

Ze względu na ładowanie aplikacji do pamięci ulotnej DSP każdorazowo po włączeniu zasilania lub restarcie procesora, konieczne jest ponowne jej załadowanie. Gdy DSP ma pracować samodzielnie, należy wykorzystać tryb *self boot*, w którym, każdorazowo po restarcie, DSP automatycznie ładuje zawartość pamięci nieulotnej. W wypadku procesorów z rodziny ADAU146x (ADAU145x) programator USBi nie ma bezpośredniego połączenia z pamięcią

EEPROM, co powoduje odmienny sposób programowania pamięci. Przed zapisem programu interfejs USBi zapisuje do DSP bootloader, który pośredniczy w zapisie EEPROM przyłączonej do interfejsu typu slave. Aby zapisać aplikację w pamięci EEPROM, należy wrócić do zakładki *Hardware Configuration*, kliknąć prawym przyciskiem myszy na blok „DSP IC1” i wybrać opcję *Self-boot Memory* → *Erase memory through DSP* i następnie *Write latest Compilation through DSP* (rysunek 23). Po zaprogramowaniu przełącznik *selfboot* należy ustawić w położenie 1=OFF i nacisnąć przycisk „reset”, co uruchomi procedurę *selfboot*, czyli wczytanie i uruchomienie aplikacji w trybie samodzielnym.

Na koniec

Przykładowa aplikacja w możliwie krótki sposób pokazuje konfigurację portów szeregowych i GPIO procesora DSP, konieczną do tworzenia samodzielnych aplikacji, i oczywiście, nie wyczerpuje wszystkich jego możliwości. Pozostaje więc dalsze samodzielne zdobywanie doświadczenia. Pomocne mogą być w tym wpisy z forum <http://ez.analog.com>. Powodzenia we własnych projektach!

Adam Tatuś, EP

REKLAMA

Klub Aplikantów Próbek

to inicjatywa redakcji „Elektroniki Praktycznej”. W kontaktach z firmami redakcja często otrzymuje do przetestowania próbki podzespołów, modułów, a nawet całych urządzeń elektronicznych. Są to zwykle najnowsze typy/modele produktów na rynku. Z chęci podzielenia się z Czytelnikami tymi próbkami zrodziła się inicjatywa pod nazwą Klub Aplikantów Próbek. Członkiem KAP staje się każdy, kto zgłosi chęć przetestowania próbki. Wykaz i krótki opis próbek, którymi dysponuje redakcja EP, można znaleźć na stronie www.ep.com.pl/KAP. Wystarczy wybrać rodzaj próbek i zwrócić się majlmem (na adres: Szef Pracowni Konstrukcyjnej grzegorz.becker@ep.com.pl) z prośbą o przesłanie bezpłatnych próbek, podając ich nazwę i adres wysyłki. Warto dopisać jaki jest plan zastosowania tych próbek. Nie jest to konieczne, ale może mieć znaczenie przy podziale próbek w przypadku większej liczby zgłoszeń. Mile widziane, choć nieobowiązkowe, jest też przystanie do redakcji EP opisu wykonanej aplikacji próbek, oczywiście po jej wykonaniu z zastosowaniem otrzymanej próbki. Autorem przysłanych opisów przyznajemy punkty, które będą im dawały pierwszeństwo przy ubieganiu się o kolejne próbki. Najciekawsze opisy aplikacji opublikujemy na forum ep.com.pl lub na łamach „Elektroniki Praktycznej”.

Dla pełnej jasności jeszcze raz podkreślamy, że próbki przekazujemy bezpłatnie i nie trzeba ich zwracać do redakcji.

Z uwagi na ograniczoną liczbę dostępnych próbek i niemałe zainteresowanie nimi, prosimy o opisanie swojego pomysłu na projekt na naszym forum internetowym, w dziale poświęconym Klubowi Aplikantów Próbek

<https://forum.ep.com.pl/viewforum.php?f=80>.

Ponadto, by zwiększyć swoje szanse na bycie wybranym do realizacji projektu w oparciu o nasze próbki, należy polubić fanpage Elektroniki Praktycznej na Facebooku (<https://web.facebook.com/ElektronikaPraktyczna>) oraz udostępnić post, w którym opisujemy rozdawane próbki. W przypadku podobnie interesujących pomysłów na projekty, będziemy uwzględniać to jako dodatkowe kryterium wyboru.



www.ep.com.pl/kap